

DIALOG(R)File 351:Derwent WPI  
(c) 2006 The Thomson Corp. All rts. reserv.

012875688    \*\*Image available\*\*

WPI Acc No: 2000-047521/200004

XRPX Acc No: N00-037016

Printing characteristics setting mechanism for laser beam printer - has  
judging unit which compares printing area coordinate information and  
printing positional information regarding character row and judges  
existence of overlap

Patent Assignee: CANON KK (CANO )

Inventor: MITSUHASHI S

Number of Countries: 002    Number of Patents: 002

Patent Family:

| Patent No   | Kind | Date     | Applicat No | Kind | Date     | Week      |
|-------------|------|----------|-------------|------|----------|-----------|
| JP 11309917 | A    | 19991109 | JP 98118739 | A    | 19980428 | 200004 B. |
| US 6535293  | B1   | 20030318 | US 99298962 | A    | 19990426 | 200322    |

Priority Applications (No Type Date): JP 98118739 A 19980428

Patent Details:

| Patent No | Kind | Lan | Pg | Main IPC | Filing Notes |
|-----------|------|-----|----|----------|--------------|
|-----------|------|-----|----|----------|--------------|

|             |   |  |    |             |  |
|-------------|---|--|----|-------------|--|
| JP 11309917 | A |  | 23 | B41J-021/00 |  |
|-------------|---|--|----|-------------|--|

|            |    |  |  |             |  |
|------------|----|--|--|-------------|--|
| US 6535293 | B1 |  |  | G06K-015/00 |  |
|------------|----|--|--|-------------|--|

Abstract (Basic): JP 11309917 A

NOVELTY - A registration unit registers coordinate information in  
printing area, based on print information generated depending upon  
printing demand indication. An area judging unit compares the printing  
area coordinate information and printing positional information  
regarding character row, and judges existence of overlap. Based on the  
decision result, the printing pattern of the character row can be  
varied. DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for  
the following: printing control system; data processor; data processing  
procedure; recording medium for storing printing characteristic setting  
program

USE - For laser beam printer.

ADVANTAGE - Enables to judge correlation with each printing object,  
thus optimum printing characteristics can be established. DESCRIPTION  
OF DRAWING(S) - The figure shows the sectional view of laser beam  
printer.

Dwg.1/25

Title Terms: PRINT; CHARACTERISTIC; SET; MECHANISM; LASER; BEAM; PRINT;  
JUDGEMENT; UNIT; COMPARE; PRINT; AREA; COORDINATE; INFORMATION; PRINT;  
POSITION; INFORMATION; CHARACTER; ROW; JUDGEMENT; EXIST; OVERLAP

Derwent Class: P75; T01

International Patent Class (Main): B41J-021/00; G06K-015/00

International Patent Class (Additional): G06F-003/12

File Segment: EPI; EngPI

?

**BEST AVAILABLE COPY**

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-309917

(43)Date of publication of application : 09.11.1999

(51)Int.Cl.

B41J 21/00

G06F 3/12

(21)Application number : 10-118739

(71)Applicant : CANON INC

(22)Date of filing : 28.04.1998

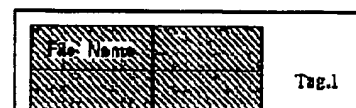
(72)Inventor : MIHASHI TOSHIYA

## (54) PRINT SYSTEM AND PRINT CONTROL METHOD, DATA PROCESSOR, DATA PROCESSING METHOD, RECORDING MEDIUM

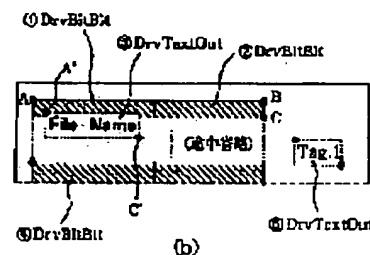
(57)Abstract:

**PROBLEM TO BE SOLVED:** To enhance convenience for user by switching the printing form of character string based on the result of a decision whether the coordinate information in the writing area of a print object is lapped over the print position information of character string or not.

**SOLUTION:** A printer driver delivers the coordinate information at an upper left point A and a lower right point B with a rule part on the surface as a rectangular image (line having thickness = rectangle) to an area information registering section where the coordinate information is processed. Similarly, the coordinate information at an upper left point A and a lower right point C of a halt tone pattern part is delivered to the area information registering section where the coordinate information is processed. When a character string of File Name is provided, for example, a decision is made whether the information of its circumscribed rectangle is lapped over the registered rectangular area or not. Character processing is performed at the writing section on the computer side if any overlap is present otherwise the character string is converted into PDL(print description language) code and processed at the writing section on the printer side.



(a)



(b)

## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-309917

(43) 公開日 平成11年(1999)11月9日

(51) Int.Cl.<sup>6</sup>

識別記号

F I

B 4 1 J 21/00

B 4 1 J 21/00

Z

G 0 6 F 3/12

C 0 6 F 3/12

B

F

審査請求 未請求 請求項の数26 O L (全 23 頁)

(21) 出願番号 特願平10-118739

(22) 出願日 平成10年(1998)4月28日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 三橋 俊哉

東京都大田区下丸子3丁目30番2号 キヤ

ノン株式会社内

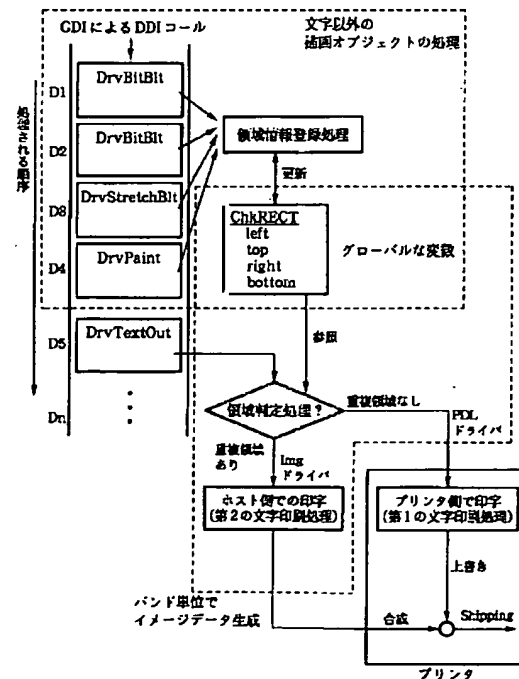
(74) 代理人 弁理士 大塚 康徳 (外2名)

(54) 【発明の名称】 印刷システムおよび印刷制御方法、データ処理装置、データ処理方法、記録媒体

(57) 【要約】

【課題】 印刷オブジェクトの内容に基づき最適な印刷処理モードを設定するためには、各描画オブジェクトとの相関関係（重なり具合）を判定する必要があるが、実際の判定作業は煩雑であり、最適な印刷処理モードの設定は困難を極めるものである。

【解決手段】 印刷要求指示に応じて印刷装置に対する印刷情報を生成し、生成された印刷オブジェクトの描画領域の座標情報を登録領域に登録する。文字列の印刷の際に登録された座標情報と文字列の印刷位置情報とを比較し、両者の重なりの有無を判定する。重複有無の判定結果に基づき文字列の印刷の処理形態を切替え、印刷オブジェクトに対して最適な印刷処理モードの設定を可能とした。



**【特許請求の範囲】**

【請求項1】 所定の通信媒体を介して印刷装置と通信可能なデータ処理装置とその印刷装置によって構成される印刷システムであって、

印刷要求指示に応じて印刷装置に対する印刷情報を生成する印刷情報生成手段と、

前記印刷情報として生成された印刷オブジェクトの描画領域の座標情報を登録領域に登録する領域情報登録手段と、

文字列の印刷の際に前記領域情報登録手段によって登録された座標情報と文字列の印刷位置情報とを比較し両者の重なりの有無を判定する領域判定手段と、  
を備え、前記判定結果に基づき前記文字列の印刷の処理形態を切替えることを特徴とする印刷システム。

【請求項2】 前記領域判定手段の判定の結果に基づき、前記印刷オブジェクトの描画領域に重複がない場合は、前記印刷装置側で展開された文字パターンを印刷する第1の文字印刷手段と、

前記印刷オブジェクトの描画領域に重複がある場合は、前記印刷データ処理装置側で展開された文字パターンをビットマップイメージとして印刷する第2の文字印刷手段の2つの文字印刷手段と、

を更に備えることを特徴とする請求項1記載の印刷システム。

【請求項3】 前記領域情報登録手段は、前記描画オブジェクトの外接矩形の領域情報を用いて登録を行うことを特徴とする請求項1記載の印刷システム。

【請求項4】 前記領域情報登録手段は、文字の下地となる可能性のある各描画オブジェクトについて、その外接矩形の座標のTop-Leftの最小値とBottom-Rightの最大値を求め前記登録領域に登録することを特徴とする請求項1記載の印刷システム。

【請求項5】 前記領域判定手段は、前記印刷情報生成手段において逐次生成される印刷情報に基づき逐次的に判定処理することを特徴とする請求項1記載の印刷システム。

【請求項6】 前記印刷情報生成手段は、1ページを複数のバンド領域に分割した単位ごとにラスター化された印刷情報を生成することを特徴とする請求項1記載の印刷システム。

【請求項7】 前記印刷情報生成手段は、更に前記領域判定手段の結果により前記複数のバンド領域間で、文字に関する印刷オブジェクトの描画領域がまたがる場合は、前記複数のバンド領域を仮想的に結合して1つの領域として拡張して処理することを特徴とする請求項1記載の印刷システム。

【請求項8】 前記領域情報登録手段および前記領域判定手段は、前記印刷情報生成手段におけるバンド領域を複数のサブエリアとして分割管理して登録、判定を行うことを特徴とする請求項1記載の印刷システム。

【請求項9】 前記領域情報登録手段は、前記印刷オブジェクトの全てを包含する最大矩形領域を定義する座標値情報を前記登録領域に保存することを特徴とする請求項1記載の印刷システム。

【請求項10】 前記登録領域は、前記印刷オブジェクトの描画領域の属性を識別するflag部と、前記描画領域が結合したときの親となる領域を識別するためのparentArea部と、

前記印刷オブジェクトの座標情報を保存するための管理領域と、

下地の領域情報を保存するための登録領域と、  
を有することを特徴とする請求項1記載の印刷システム。

【請求項11】 前記領域判定手段は、更に、PDLとして処理した文字列の領域と、その後に入力された前記文字列の下地となる印刷オブジェクトの座標情報とが重複すると判定した場合は、前記重複を識別するための画像保証フラグを立て、前記画像保証フラグが立った以降は、全ての印刷オブジェクトを前記第1の文字印刷手段により印刷処理することで、前記印刷オブジェクトの描画順序の繰り上がりによる印刷結果の不整合を解消することを特徴とする請求項1記載の印刷システム。

【請求項12】 前記バンド領域内で、PDLとして処理した文字列の印刷情報を前記データ処理装置側に保持する保持手段を更に備え、

前記領域判定手段により、前記PDLとして処理した文字列の領域と、その後に入力された前記文字列の下地となる印刷オブジェクトの座標情報とが重複すると判定された場合は、前記保持手段に保持されている文字列の印刷情報を呼び出し、前記文字列の印刷情報に基づき、前記データ処理装置側で描画メモリに展開し、前記登録領域に登録された描画領域の座標情報を前記文字列の印刷情報をも対象として更新することで、前記印刷オブジェクトの描画順序の繰り上がりによる印刷結果の不整合を解消することを特徴とする請求項1記載の印刷システム。

【請求項13】 印刷要求指示に応じて印刷装置に対する印刷情報を生成する印刷情報生成工程と、

前記印刷情報として生成された印刷オブジェクトの描画領域の座標情報を登録領域に登録する領域情報登録工程と、

文字列の印刷の際に前記領域情報登録工程によって登録された座標情報と文字列の印刷位置情報とを比較し両者の重なりの有無を判定する領域判定工程と、

を備え、前記判定結果に基づき前記文字列の印刷の処理形態を切替えることを特徴とする印刷制御方法。

【請求項14】 前記領域判定工程の判定の結果に基づき、前記印刷オブジェクトの描画領域に重複がない場合は、前記印刷装置側で展開された文字パターンを印刷する第1の文字印刷工程と、

前記印刷オブジェクトの描画領域に重複がある場合は、前記印刷データ処理装置側で展開された文字パターンをビットマップイメージとして印刷する第2の文字印刷工程の2つの文字印刷工程と、  
を更に備えることを特徴とする請求項1記載の印刷制御方法。

【請求項15】 前記領域情報登録工程は、前記描画オブジェクトの外接矩形の領域情報を用いて登録を行うことを特徴とする請求項1記載の印刷制御方法。

【請求項16】 前記領域情報登録工程は、文字の下地となる可能性のある各描画オブジェクトについて、その外接矩形の座標のTop-Leftの最小値とBottom-Rightの最大値を求め前記登録領域に登録することを特徴とする請求項1記載の印刷制御方法。

【請求項17】 前記領域判定工程は、前記印刷情報生成工程において逐次生成される印刷情報に基づき逐次的に判定処理することを特徴とする請求項1記載の印刷制御方法。

【請求項18】 前記印刷情報生成工程は、1ページを複数のバンド領域に分割した単位ごとにラスター化された印刷情報を生成することを特徴とする請求項1記載の印刷制御方法。

【請求項19】 前記印刷情報生成工程は、更に前記領域判定工程の結果により前記複数のバンド領域間で、文字に関する印刷オブジェクトの描画領域がまたがる場合は、前記複数のバンド領域を仮想的に結合して1つの領域として拡張して処理することを特徴とする請求項1記載の印刷制御方法。

【請求項20】 前記領域情報登録工程は、前記印刷オブジェクトの全てを包含する最大矩形領域を定義する座標値情報を前記登録領域に保存することを特徴とする請求項1記載の印刷制御方法。

【請求項21】 前記登録領域は、前記印刷オブジェクトの描画領域の属性を識別するflag部と、前記描画領域が結合したときの親となる領域を識別するためのparentArea部と、前記印刷オブジェクトの座標情報を保存するための管理領域と、下地の領域情報を保存するための登録領域と、を有することを特徴とする請求項1記載の印刷制御方法。

【請求項22】 前記領域判定工程は、更に、PDLとして処理した文字列の領域と、その後に入力された前記文字列の下地となる印刷オブジェクトの座標情報とが重複すると判定した場合は、前記重複を識別するための画像保証フラグを立て、前記画像保証フラグが立った以降は、全ての印刷オブジェクトを前記第1の文字印刷工程により印刷処理することで、前記印刷オブジェクトの描画順序の繰り上がりによる印刷結果の不整合を解消することを特徴とする請求項1記載の印刷制御方法。

【請求項23】 前記バンド領域内で、PDLとして処理した文字列の印刷情報を前記データ処理装置側のメモリに保持する保持工程を更に備え、

前記領域判定工程により、前記PDLとして処理した文字列の領域と、その後に入力された前記文字列の下地となる印刷オブジェクトの座標情報とが重複すると判定された場合は、前記メモリに保持されている文字列の印刷情報を呼び出し、前記文字列の印刷情報に基づき、前記データ処理装置側で描画メモリに展開し、前記登録領域に登録された描画領域の座標情報を前記文字列の印刷情報をも対象として更新することで、前記印刷オブジェクトの描画順序の繰り上がりによる印刷結果の不整合を解消することを特徴とする請求項1記載の印刷制御方法。

【請求項24】 所定の通信媒体を介して印刷装置に印刷情報を出力するデータ処理装置であって、印刷装置に対する印刷情報を生成する印刷情報生成手段と、前記生成された印刷情報の描画領域の座標情報を登録領域に登録する領域情報登録手段と、前記座標情報を登録する際、前記描画領域を包含する最大領域として座標情報を更新する領域情報更新手段と、文字列の印刷の際に前記領域情報登録手段によって登録された座標情報と文字列の印刷位置情報とを比較し両者の重なりの有無を判定する領域判定手段と、を備えることを特徴とするデータ処理装置。

【請求項25】 印刷装置に対する印刷情報を生成する印刷情報生成工程と、前記生成された印刷情報の描画領域の座標情報を登録領域に登録する領域情報登録工程と、前記座標情報を登録する際、前記描画領域を包含する最大領域として座標情報を更新する領域情報更新工程と、文字列の印刷の際に前記領域情報登録手段によって登録された座標情報と文字列の印刷位置情報とを比較し両者の重なりの有無を判定する領域判定工程と、を備えることを特徴とするデータ処理方法。

【請求項26】 印刷装置に対する印刷情報を生成する印刷情報生成工程と、前記生成された印刷情報の描画領域の座標情報を登録領域に登録する領域情報登録工程と、前記座標情報を登録する際、前記描画領域を包含する最大領域として座標情報を更新する領域情報更新工程と、文字列の印刷の際に前記領域情報登録手段によって登録された座標情報と文字列の印刷位置情報とを比較し両者の重なりの有無を判定する領域判定工程と、をコンピュータに実行させるためのプログラムを記録したことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、データ処理装置

(ホスト・コンピュータ)からの印刷情報を受けて動作する印刷装置と、その印刷装置のためにホスト・コンピュータ側で印刷情報を生成するプリンタ・ドライバ、デバイス・ドライバなどと呼ばれるソフトウェア・モジュール群をあわせた印刷データ処理装置を含む印刷システムおよび印刷制御方法、データ処理装置、データ処理方法、記録媒体に関するものである。

【0002】

【従来の技術】従来の印刷システムにおける印刷制御方法および処理の流れについてまず説明する。

【0003】一般的には、セントロニクスインターフェースといったパラレル通信手段やネットワーク通信手段を介してホスト・コンピュータと印刷装置とが接続され印刷システムを構成している。

【0004】ホスト・コンピュータ側では、ワードプロセッサや表計算のようなアプリケーションソフトウェア(以下アプリと略称)がWindows(米国Microsoft社の登録商標)のようないわゆる基本ソフトの上で動作している。このアプリにおいて印刷を行う場合は、そのプログラム内から、基本ソフトが提供するいくつかのサブシステムのうちのグラフィック・サブ・システムの機能を用いて行う。

【0005】このグラフィック・サブ・システムは、例えばWindowsでは、GDI(Graphic Device Interface)と呼ばれておりディスプレイやプリンタに対する画像情報の処理を司っている。

【0006】このGDIは、ディスプレイやプリンタといった各デバイス毎の依存性を吸収するためにデバイスドライバと呼ばれるモジュールを動的にリンクし、それぞれのデバイスに対する出力処理を行う。プリンタに対するこのモジュールはプリンタ・ドライバと呼ばれる。このプリンタ・ドライバでは、その能力や機能などに応じてあらかじめデバイスドライバに実装することが決められているDDI(Device Driver Interface)と呼ばれる関数群を用意する必要がある。アプリのAPI(Application Programming Interface)コールをGDIがデバイスドライバ用にデータ変換を行い、このDDI関数群が適宜GDIからコールされ所定の印刷処理が実行されるような仕組みになっている。GDIでは、このようにプリンタ・ドライバを介してアプリからの印刷要求をシーケンシャルに処理する。このプリンタドライバの処理系は、PDLタイプとイメージタイプの2つに大別できる。

【0007】PDLタイプは、印刷装置側にPDL(Printer Description Language)と呼ばれる制御コマンドを処理可能なコントローラを搭載するもの向けで、システムから渡されるDDIをPDLのコマンドに変換する処理を中心に行う。

【0008】一方、イメージタイプは、印刷装置側では高度な描画処理は行わず、プリンタドライバ側で印刷イ

メージの展開を行い、そのイメージを印刷装置に送って印刷する。

【0009】例えば、文字の印刷処理については、PDLタイプでは、印刷装置に内蔵するフォントを利用して印刷をおこなったり、あらかじめプリンタドライバ側で必要に応じて展開した文字パターンに固有のIDを割り振り、まずは文字パターンを印刷装置側に登録し、続いて印刷位置情報とIDによって文字印刷を行うような仕組みを持っている。

【0010】イメージタイプでは、プリンタドライバ側においてバンド単位もしくはフルページの描画メモリを持ち、そこに文字を展開する。1バンド分もしくはフルページ分の描画処理が終了すると、印刷装置に対してイメージデータとして描画メモリの内容が転送され所定の画像を印刷出力する。

【0011】従来は、このPDLタイプとイメージタイプはひとつのプリンタドライバの中で排他的に利用されるものがあつた。どちらを利用するかはユーザが印刷すべきドキュメントの内容によって、ユーザ自身が判断し、プリンタドライバのインターフェース(以降、UIと略称)から設定するのが一般的であつた。

【0012】

【発明が解決しようとする課題】しかしながら上記従来例では次のような問題が存在する。

【0013】(1)ドキュメントの内容によってユーザ自身が最適な処理モード(PDL or イメージ)を判断するのは困難な場合がある。

【0014】これは、文字が多いためにPDLの方が高速処理できると限らないためである。例えば、用紙の全面に写真のようなイメージがあつて、その上に文字が書かれているような場合(図3)は、イメージドライバで処理した方が高速になる可能性が高い。つまり、描画する面積、文字に下地があるかどうか、複数ページのドキュメントかどうか(Font Cacheがヒットするかどうか)によって印刷結果を得るまでの処理時間はドライバの動作タイプによって大きく異なるものである。

【0015】図3に示すような印刷の場合は、全面がイメージであり、このような場合に文字分離処理を行い、文字を別途切り出してPDLとして処理しても、その文字をPDLと処理する分のコマンドや文字データが増えることで、全体としての印刷データは増加し、イメージ描画と文字印刷という2つ処理を行う必要があるため処理量の観点からもオーバーヘッドが大きい。

【0016】このような全面イメージの場合は、ホスト側で展開したイメージの上に文字もイメージとして貼り付け、全体をイメージとしてプリンタに送出するといったイメージタイプのプリンタドライバの方が優位であるということである。

【0017】(2)ユーザに最適な処理モードを選択させるということは、プリンタドライバのUIの複雑化を

招き、かえってユーザーの混乱を招く恐れがある。例えば、キャノン製プリンタドライバでは、印刷条件の異なる設定を図4に示すようなアイコン化することによってユーザによる条件設定の負担を軽減しているが、それでもさまざまな印刷モードが増えるとユーザの混乱を招く恐れがあることは否めない。

【0018】(3) 上記ユーザによる選択時の問題を解決する方法として、DDIの情報を一旦ファイルとして格納し、その情報をプリンタドライバ自身が分析して最適な処理モードを厳密に確定する方法も考えられているが、一旦ファイル化するために処理時間がかかり結果的にパフォーマンスの低下を招く恐れがある。

【0019】(4) 最適な処理モードを厳密に確定するためには、各描画オブジェクトとの相関関係(重なり具合)を判定する必要がある、描画オブジェクトが非常に多い場合はその検出処理がかえってオーバーヘッドとなって期待する効果が得られない恐れがある。

【0020】(5) 文字とその他の描画オブジェクトの重なり具合を厳密に検査する場合、描画メモリ上でパターンの重なりを検証するとピクセル単位の処理になり、そのピクセル単位の検出処理がかえってオーバーヘッドとなって期待する効果が得られない恐れがある。

【0021】例えば図5に示すようにABCという文字とハッチングパターン付きの楕円と1本の斜め線が密接して描画されているような場合、それぞれの重なり具合を厳密に検証することは難しい。このような場合、従来は点線枠で示すような文字列を取り囲むFONT Boundary Boxと呼ばれる情報によって判別するのが一般的である。但し、このような判別の場合、例えば図5に示した例では実際には各オブジェクトの描画パターン同士の重なりは無いにもかかわらず、領域判定の結果は重なり(ハッチング部分と点線枠との重なり部分を判断して)有りと判定されることもある。

【0022】PDLタイプとイメージタイプのドライバは、一般的に従来は印刷ジョブ単位で切り替えているが、印刷データの内容に応じて各オブジェクト毎に両者を使い分けるハイブリッド方式も考えられる。しかしながら、そのようなハイブリッド方式の構成にした場合、次のような問題が考えられる。

【0023】(6) ハイブリッド方式では、PDLタイプとイメージタイプの両方の構成を兼ね備えるが、例えば、イメージタイプの構成ではメモリの使用効率の観点からバンディング方式をとる場合が多い。このような構成で文字印刷を行う場合、文字はPDLで処理するものであっても、バンド間にまたがる文字は、文字印刷のDDIコールが2回発生することになる。つまり図6に示すように文字の上半分はクリッピングをかけて最初のバンドに展開するようなDDIコールが発生し、文字の下半分はやはりクリッピングをかけて次のバンドに文字の下半分を描画するといった具合である。この処理形態

は、DDIコールのレベルでそのような処理工程が発生しているため、イメージタイプであろうとPDLタイプであろうと同様の処理形態となる。このようにバンド間にまたがる文字の処理は、1文字の印刷に4工程(①バンドnのクリッピング領域の設定、②文字パターンの展開、③バンドn+1のクリッピング領域の設定、④文字パターンの展開)の処理を必要とするため処理速度の低下を招いている。

【0024】

【課題を解決するための手段】まず、前述の課題(1)に対しては、印刷する文字に下地があるかどうか領域判定の際のポイントとなる。

【0025】また、課題(2)に対しては、ユーザ側に選択させるのではなく、原則的にはドライバ自身が判断し最適な処理方法で印刷処理を行うという自律的な判断機構を持つ構成であることが望ましい。

【0026】さらに課題(3)に対しては、1パスめでDDI情報のファイル化を行い、そのファイルの情報を調べて判断するといったnパス方式をやめ、DDI情報が来るたび毎に逐次的(即時的)に判断を行う方式とすることが望ましい。

【0027】また、課題(4)に関しては、本発明の主旨が印刷パフォーマンスの改善でもあることより、ドットレベルでの重なり具合の厳格さより、処理速度を優先して考えることが望ましく、文字列の外接矩形情報を用いて判断する方式が望ましい。

【0028】さらに、課題(5)に対しては、文字分離処理によって切り出した文字は下地がないという条件に限定できれば、PDLで印刷時はクリッピングの必要性が無くなる上に2回めの印刷処理も省くことが可能である。

【0029】本発明では、上記従来の課題に対して以上のような対応方法を考慮することにより、主に次のような構成によって問題を解決を図るものである。

【0030】つまり、所定の通信媒体を介して印刷装置と通信可能なデータ処理装置とその印刷装置によって構成される印刷システムは、印刷要求指示に応じて印刷装置に対する印刷情報を生成する印刷情報生成手段と、前記印刷情報として生成された印刷オブジェクトの描画領域の座標情報を登録領域に登録する領域情報登録手段と、文字列の印刷の際に前記領域情報登録手段によって登録された座標情報と文字列の印刷位置情報とを比較し両者の重なりの有無を判定する領域判定手段とを備え、前記判定結果に基づき前記文字列の印刷の処理形態を切替えることを特徴とする。

【0031】また、印刷制御方法は、印刷要求指示に応じて印刷装置に対する印刷情報を生成する印刷情報生成工程と、前記印刷情報として生成された印刷オブジェクトの描画領域の座標情報を登録領域に登録する領域情報登録工程と、文字列の印刷の際に前記領域情報登録工程

によって登録された座標情報と文字列の印刷位置情報とを比較し両者の重なりの有無を判定する領域判定工程とを備え、前記判定結果に基づき前記文字列の印刷の処理形態を切替えることを特徴とする。

【0032】但し、本発明は原則として、次のような前提があることを条件として成り立つものであり、この限られた前提条件下で正しい印刷結果を保証するものである（但し、あくまでも原則であり、原則が崩れる場合への対応を実施形態3で説明している）。その前提条件とは、文字の下地となる描画オブジェクトは、その文字を取り囲むように存在し、印刷処理時には文字印刷DDIより先に下地となる描画オブジェクトのDDIがコールされる。また、その文字の上に論理演算を行うような描画オブジェクトは描画されないというものである。

【0033】例えば、図7(a)に示すような印刷を行う場合、“File Name”という文字列は下地を含む。このような場合の文字はPDLで処理すべき文字として切り出さずイメージタイプの処理とする。その際の判別方法を図7(b)を用いて簡単に説明する。このような下地を含む文字列の下地は、一般的な印刷文書においては、その文字列を取り囲むように存在する。そのような場合、システムがプリンタドライバに渡す印刷指示も描画領域の情報から順に処理されるのが一般的である。

【0034】つまりこの図7の場合、まず下地となる表の罫線部分（直線）が矩形イメージとしてDrvBitBlt(Φ)というDDIコールによってプリンタドライバに渡されて来る。この時点でプリンタドライバはまず、その描画オブジェクト（この場合は矩形）の左上の点（A点）と右下の点（B点：直線の矩形イメージなので直線の右側端点部）の2の座標情報を所定の領域情報記憶手段に登録する。

【0035】次に表の網掛けパターン部である部分（矩形）がDrvBitBlt(Φ)というDDIコールによってプリンタドライバに渡されて来る。ここでもまず領域情報登録手段によってその矩形領域の左上の点（A点）と右下の点（C点）の2の座標情報が所定の領域情報記憶手段に登録されるが、ここで既に登録してある領域情報と比較し、より左上方向（Top Left）に小さな値、より右下方向（Bottom Right）に大きな値を登録するようにする。つまり領域情報記憶手段が管理する領域情報は、この場合では常に2点としておく。

【0036】このように描画オブジェクトごとにその領域情報を参照して、常にそれらの描画オブジェクトを含む最大の矩形領域を把握しておくようにする。このような状態で、次に文字列が来た場合、まず、文字列を含む外接矩形の情報と、それまでに領域情報登録手段によって登録した矩形領域との重なりの有無を調べる。

【0037】ここで領域に少しでも重なりが存在すれば、その文字列は下地を含むものとして判断し、イメージタイプとして処理する。例えば“File Nam

e”(Φ)はA'とC'とで定義される矩形領域と、Φの網掛け部分とは重複する。

【0038】一方この領域判定で重なり無しと判定された場合（例えば図7の“Tag. 1”という文字列部分）は、その文字列はPDLタイプで処理することで、常に最適な処理手段を即時的に判断することがなるといったものである。

【0039】以上が本発明における問題解決手段の動作原理であり、このような問題解決手段によって前述の従来の課題を解決し、印刷処理にかかる時間を短縮可能し、ユーザによる動作モード判断からプリンタドライバでの自律的な動作モード判断に移行することによって煩雑なユーザインタフェース（UI）設定を無くしユーザにとっての使い勝手を向上するものである。

【0040】

【発明の実施の形態】<実施形態1>本実施形態の構成を説明する前に、本実施形態を適用する印刷システムの印刷を担うレーザービームプリンタ（以下「LBP」と記述）の構成について図1を参照しながら説明する。

【0041】図1は、LBPの構成を示す断面図である。図において、100はLBP本体であり、外部に接続されているホスト・コンピュータなどから供給されるプリントデータ（文字コードやイメージデータ等）及び制御コードから成る印刷情報を入力して記憶するとともに、それらの情報に従って対応する印刷イメージパターンや印刷イメージ等を作成し、記憶媒体である記録紙上に像を形成する。

【0042】120は操作のためのスイッチおよびLED表示器等が配されているオペレーション・パネル、110はLBP本体100の制御およびホスト・コンピュータから供給される文字情報等を解析し印刷処理を行うフォーマット制御部である。このフォーマット制御部110において展開された印刷情報は、対応するパターンのビデオ信号に変換されレーザードライバ131に出力される。レーザードライバ131は半導体レーザー141を駆動するための回路であり、入力されたビデオ信号に応じて半導体レーザー141から発射されるレーザー光142をオン・オフに切り替える。

【0043】レーザー光142は回転多面鏡143で左右方向に振らされて静電ドラム144上を走査露光する。これにより、静電ドラム144上には文字パターンの静電潜像が形成されることになる。この潜像は、静電ドラム144周囲に配設された現像ユニット145により現像された後、記録紙に転写される。この記録紙にはカットシートを用い、カットシート記録紙はLBP100に装着した用紙カセット146に収納され、給紙ローラ147及び搬送ローラ148と149とにより、装置内に取り込まれて、静電ドラム144に供給され、記録紙に転写され、画像を形成する。

【0044】また、LBP本体100には、図示しない

カードスロットまたはFlashROMのようなメモリを装着するためのメモリスロットを1個以上備え、内蔵フォントに加えてオプションフォントカード、言語体系(コマンド体系)の異なる制御カード(エミュレーションカード)を接続できるように構成されている。

【0045】なお、本発明を適用可能なプリンタは、LBPに限られるものではなく、カラーLBPであってもインクジェットプリンタをはじめとする他のプリント方式のプリンタでも良いことは言うまでもない。

【0046】印刷装置100において、フォーマット制御部110は、ホスト・コンピュータとの接続手段であるところのインタフェース(I/F)部111と、受信データ等を一時的に保持管理するための受信バッファ1121、送信データ等を一時的に保持管理するための送信バッファ1122、印刷データの解析を司るPDL解析部113、印刷制御処理実行部114、描画処理実行部115、ページメモリ116等より構成されている。

【0047】インタフェース(I/F)部111は、ホスト・コンピュータ200との印刷データの送受信を行う通信手段あり、通信プロトコルとしてIEEE-1284に準拠した通信を可能とするものである。このインタフェース部111を通して受信した印刷データは、そのデータを一時的に保持する記憶手段である受信バッファ1121に逐次蓄積され、必要に応じてPDL解析部113または描画処理実行部115によって読み出され処理される。PDL解析部113は、各PDLコマンド体系や印刷ジョブ制御言語に準じた制御プログラムにより構成されており、このPDL解析部113で解析されたコマンドは、文字印刷、図形、イメージなどの描画に関する印刷データの解析結果は、描画処理実行部115に指示を与えて処理し、給紙選択やリセット命令などの描画以外のコマンドは、印刷制御処理実行部114に指示を出し処理する。

【0048】描画処理実行部115では、文字やイメージの各オブジェクトをページメモリ116に逐次展開して行く。また符号化されたイメージデータを復号化してページメモリに展開する際には、この描画処理実行部115においてデータ復号化処理がなされる。この展開処理とプリンタエンジンへのビデオ信号の SHIPPING の追いかけて、つまりバンディング制御によってページメモリ116は管理されていてもよいし、十分なメモリがある場合は、1ページ分が展開可能な領域を確保してもよい。なお、一般的に、フォーマット制御部110は、中央演算処理装置(CPU)、リードオンリーメモリ(ROM)、ランダムアクセスメモリ(RAM)などを用いたコンピュータ・システムによって構成されている。また、各部の処理は、マルチタスクモニタ(リアルタイムOS)のもとで、タイムシェアリングに処理される構成であっても良いし、各機能ごとに専用のコントローラ・ハードウェアを用意して独立して処理される構成

であってもかまわない。

【0049】オペレーション・パネル120は、印刷装置の各種状態を設定・表示するためのものである。出力制御部130は、ページメモリ116の内容をビデオ信号に変換処理し、プリンタエンジン部140へ画像転送を行う。プリンタエンジン部140は受け取ったビデオ信号を記録紙に永久可視画像形成するための印刷機構部であり、図1において前述したものである。給紙カセット146は着脱可能な用紙格納装置である。

【0050】以上、印刷装置100について説明したが、次にホスト・コンピュータ200を含む本実施形態の印刷システムの全体構成について説明を加える。

【0051】なお、本発明の機能が実行されるのであれば、単体の機器であっても、LAN等のネットワークを介して処理が行われる複数の機器からなるシステムであっても本発明を適用可能である。

【0052】図2において200は印刷システムにおけるデータ処理装置であるところのホスト・コンピュータであり、プリントデータ及び制御コードから成る印刷情報を印刷装置100に出力するものである。ホスト・コンピュータ200は、入力デバイスであるところのキーボード210やポインティングデバイスであるところのマウス211と、表示デバイスであるディスプレイ・モニタ220を合わせた一つのコンピュータ・システムとして構成されている。ホスト・コンピュータ200は、Unix、Windowsなどの基本OSによって動作しているものとする。ホスト・コンピュータ側について、本発明に関する機能的な部分にのみ注目し、基本OS上での機能を大きく分類すると、アプリケーション201、画像情報処理手段であるところのグラフィック・サブ・システム202、データ格納手段、印刷データ格納制御手段および印刷装置との通信手段を含むスプール・サブ・システム203に大別される。

【0053】アプリケーションソフトウェア201は、例えば、ワープロや表計算などの基本ソフトウェア上で動作する応用ソフトウェアを指すものである。グラフィック・サブ・システム202は、基本OSの機能の一部であるGraphic Device Interface(以後、GDIと記す)2021とそのGDIから動的にリンクされるデバイスドライバであるところのプリンタ・ドライバ2022によって構成されている。

【0054】このプリンタドライバと呼ばれる部分が本発明における印刷情報生成手段である。スプール・サブ・システム203は、グラフィック・サブ・システム202の後段に位置するプリンタ・デバイスに特有のサブ・システムであり、データ格納手段であるところのスプールファイル(実態はハードディスク)2031などから構成されるものである。基本OSの種類によって、上述したこれらの名称や機能的な枠組みは若干異なる場合があるが、本発明で言う各技術的手段が実現できるモジ

ジュールであれば、それらの名称や枠組みは本発明にとってあまり大きな問題ではない。例えば、スプーラやスプールファイルと呼ばれるものは、別のOSにおいてプリント・キューと呼ばれるモジュールに処理を組み込むことによっても実現可能である。

【0055】なお一般的に、これらの各機能モジュールを含むホスト・コンピュータ200は、中央演算処理装置(CPU)、リードオンリーメモリ(ROM)、ランダムアクセスメモリ(RAM)、ハードディスクドライブ(HDD)、各種入出力制御部(I/O)などのハードウェアのもとで、基本ソフトと呼ばれるソフトウェアがその制御を司り、その基本ソフトの元で、それぞれの応用ソフト・サブ・システム・プロセスが機能モジュールとして動作するようになっている。

【0056】以上、図1、図2をもとに本発明を構成する各構成要素について説明をしたが、次に本発明の動作について、図8を用いて処理の流れの概要を説明する。

【0057】図8は、本発明の処理の流れを示す概念図である。例えば図7(a)に示すような印刷を行う場合を例に取り処理の流れを説明する。図7(a)は、表計算ソフトでよく利用されるような表とその表の番号を示す文字列とから構成されているデータの最初のバンド領域の印刷イメージである。

【0058】まず下地となる表の罫線部分(直線)が矩形イメージとしてDrvBitBlt(D1)というDDIコールによってプリンタドライバに渡されて来る。この時点でプリンタドライバはまず、その描画オブジェクト(この場合は矩形。直線に見えるが太さのある直線=矩形)の左上の点(A点)と右下の点(B点)の2つの座標情報を領域情報登録処理部に渡し処理する。領域情報登録処理部は、本発明の領域情報登録手段であり、プリンタドライバ内のソフトウェアとして存在し、渡された矩形情報(Top-Left, Bottom-Right)を所定の領域情報記憶領域に登録する。

【0059】この領域情報記憶領域はプリンタドライバ内のグローバルな変数として存在し、バンドが更新されるたびにこの領域も初期化される。D1の描画オブジェクトはバンド内で最初の描画オブジェクトであるため、D1の領域情報がそのまま登録される。これ以降の領域登録は、その都度、それまでに登録された情報との比較によって登録が管理されることになる。領域情報登録処理は、領域管理用のグローバルな変数領域であるChkRECTを参照、更新することで行われる。

【0060】具体的には、図9に示すプログラムのように、対象とする描画オブジェクトの領域TrgRECTとChkRECTを比較し、左上の座標値がより小さい方と、右下の座標値がより大きな方の値を求め、その値で、ChkRECT変数を更新していく。つまり、対象となるいくつかの描画オブジェクトが含まれる最大の矩形領域を算出するものである。以上のような領域登録処

理を行いながら、文字印刷のDDIが来る直前までのDDIの領域情報をチェックしておく。

【0061】例えば、D1の次の描画オブジェクトとしては、表の網掛けパターン部である部分(矩形)がDrvBitBlt(2)というDDIコールによってプリンタドライバに渡されて来る。ここでもまず領域情報登録手段によってその矩形領域の左上の点(A点)と右下の点(C点)の2の座標情報を領域情報処理部に渡し処理する。このような処理の流れで文字の下地となる可能性を持ついくつかの描画オブジェクトを処理した状態で、次に文字列が来た場合、まず、文字列を含む外接矩形の情報と、それまでに領域情報登録手段によって登録した矩形領域との重なりの有無を調べる領域判定処理を行う。

【0062】つまり先に登録・更新したChkRECT変数に格納されている領域に少しでも重なるかどうかといった領域判定処理を行う。具体的には、この領域判定処理は図10に示すようなプログラムによって実現される。

【0063】次に、文字印刷用DDIの処理部をより具体的なプログラムコードとして示したものが図11である。前述の領域判定処理の結果、処理しようとする描画オブジェクトが、ChkRECTに少しでも重なるならば、その文字印刷処理は、ホストコンピュータ側の描画処理実行部によって処理する(第2の文字印刷処理)。一方、ChkRECTと重なる部分が無い場合は、その文字印刷処理は、プリンタドライバ内でPDLコードに変換され、プリンタ側の描画処理実行部で実際の印刷処理が行われるように振り分けられる(第1の文字印刷処理)。

【0064】それぞれのデータはまず、ホスト側で描画処理の実行がされたバンド分のイメージデータがプリンタ側の描画メモリに展開され、その描画メモリに対してPDLコードとして送られてきた文字データを印刷処理することになる。但し、PDLコードとして処理する文字パターンとホスト側で描画処理の実行がされた領域との重なりはないため、処理順序は逆になっても最終的な印刷イメージの出力結果は変わらないので問題はない。

【0065】本実施形態によれば、プリンタドライバは常に最適な処理手段を自律的かつ即時的に判断することを可能とするものである。

【0066】<実施形態2>前述の実施形態1では、本発明の基本的な処理の仕組みを中心に説明したが、本発明をより実用的な領域で利用するためには、次に述べるような実施形態を取ることがより望ましい。実施形態1では、図12のような印刷を処理する場合に期待した効果が得られない場合があることが判明しているためである。つまりこのような文書中心のドキュメントでかつ文字の下地となるようなパターンやイメージが存在しない場合であれば、通常はPDLタイプで処理した方が高速

である。

【0067】しかしながら、デザイン的な理由などにより、ドキュメントの左右に縦線が2本引かれているような場合に、実施形態1の領域判別方法を適用すると、2本の縦線に挟まれた文字データ部分もすべてイメージタイプの処理がなされることになってしまう。この場合、領域登録手段によって登録される領域がA線とB線の左右端の座標となるため、文字印刷時の領域判定で実際にはパターン同士の重なりが無いにもかかわらず、常に重なり有りと判定されてしまうためである。

【0068】結果的にドキュメント全面がイメージデータとなってしまう、プリンタに送出するデータ量の増大、さらに複数ページのドキュメントの場合は、文字パターンの再利用(Font cacheにヒットするかどうか)が効かないため、印刷スループットが上がらないという問題が発生する。実施形態2は以上のような問題を想定して、より実用的な実施形態を示すものである。

【0069】実施形態2では、バンド内の領域を図13に示すような10個の領域に分けて領域管理をすることが特徴である。こうすることによって、図12のような印刷の場合に対しても適切に文字分離処理が可能となる。以下では実施形態2における処理の流れを、図14に示すフローチャートを用いて説明する。

【0070】図14は、本発明の印刷システムにおけるホスト側の処理であるプリンタドライバの処理の流れを示すものであり、S1401～S1410は各ステップを示す。このプリンタドライバは、イメージの展開に関してはバンディング方式の構成を取っている。

【0071】バンディング方式とは、1ページをいくつかに分割して、分割した単位ごとにラスターライズされた印刷情報を生成する方式をいう。

【0072】まず、ステップS1401では、各バンドにおける最初のDDI処理かどうかを確認する。最初の処理である場合はステップS1402において、そのバンド処理内で有効となる変数として、サブエリア管理用配列ChkArea[]の初期化処理を行う。サブエリア管理用配列は、初期化状態では、図18に示すような値になっている。ChkArea[]変数は、SUBAREA型の変数であるが、このSUBAREA型は図17のような形で定義されている。

【0073】すなわち、分割された印刷オブジェクトの描画領域が結合しているか、独立であるか、未登録であるかを識別するためのflag部と、描画領域が結合したときの親となる領域を識別するためのparentArea部と、印刷オブジェクトの座標情報を保存するための管理領域と、下地の領域情報を保存するための登録領域とから構成される。

【0074】次にステップS1403では、文字印刷のDDIかどうかを確認する。文字印刷のDDIでない場合は、ステップS1404に進み、その描画オブジェク

トの外接矩形の領域情報をDDIの各パラメータから抽出する。さらにステップS1405に進み、その矩形領域情報を用いて領域登録処理を実行する。不図示ではあるが、バンド内の最初の描画オブジェクトに関しては無条件にその領域が登録されることになる。領域登録処理の詳細は後述する。領域を登録した後は、各描画オブジェクトに適した描画処理を行う。ステップS1406における描画処理は既存のプリンタドライバと同じものである。以上、文字以外の描画オブジェクトに対する処理の流れを説明したが、次にステップS1403において、文字印刷のDDIであった場合の処理について説明する。

【0075】文字印刷のDDIであった場合は、まずステップS1407に進み、文字列の外接矩形情報をDDIの各パラメータから抽出する。ここでは、Font BoundaryBoxと呼ばれる文字の外接矩形情報は、グラフィックサブシステムがあらかじめその値を計算し、DDIのパラメータとしてプリンタドライバに渡して来るものとする。次にその文字列の矩形領域情報と、それまでに登録された文字以外の描画オブジェクトの領域情報との比較処理、つまり領域判定処理をステップS1408にて行う。領域判定処理の詳細については後述する。

【0076】ステップS1408の判定処理の結果、重なり有りと判定された場合は、ステップS1409にてイメージタイプの印刷処理つまりホスト側の描画メモリに対して文字パターンの展開処理を行う。

【0077】一方、重なり無しと判定された場合は、PDLタイプの印刷処理つまりホスト側からは書体選択命令や文字コードをパラメータとする文字印刷命令などを生成し、そのPDL命令によってプリンタ側の描画メモリに対して文字パターンの展開処理を行うように動作する。以上、本発明におけるプリンタドライバの処理の大きな流れについて説明したが、次に本発明の処理のポイントとなる領域登録処理と領域判定処理について詳細に説明を加える。

【0078】図15は、領域登録処理の処理の流れを示すフローチャートであり、S1501～S1506は各ステップを示す。ステップS1501では、この領域登録処理に引数として与えられた矩形領域が10個に分割したサブエリア(図13参照)の1つの領域内に収まるかどうかの判別を行う。

【0079】例えば図19に示すような直線の描画の場合は、サブエリア $\Phi$ の領域内に収まっているので、この場合はステップS1504に進む。ステップS1504では、対象となるサブエリアが後述する結合フラグの立った領域かどうかを見て処理を切り替える。

【0080】結合領域である場合は(ステップS1504-Yes)、ステップS1505に進み、そうでない単独のサブエリアである場合は(ステップS1504-No)、ステップS1506に進む。

【0081】ステップS1501において1つのサブエリア内に収まらず、複数のサブエリアにまたがるような場合はステップS1502に進み、どのサブエリア番号に該当するかを求める。そして、ステップS1503において、それら該当するサブエリアの情報を結合して、領域を拡張した新たなサブエリアを構成する。

【0082】この領域拡張されたサブエリアも、サブエリア管理用配列ChkArea[]変数によって管理されており、これらサブエリアの構成情報は、エリアの結合処理という部分で処理される。エリアの結合処理については後述する。

【0083】いくつかのサブエリアが集まって領域を広げた新たなサブエリアができるが、その際ステップS1505として、もっとも左上のサブエリアが親サブエリアとなって新たなサブエリアの領域情報を管理することになる。親以外は子となり、子サブエリアの管理用配列には、結合されたことを示すフラグが立ち、さらに親となるサブエリアのインデックス番号が格納される。子サブエリアでは、領域拡張された新たなサブエリアの領域情報は管理せず、すべて親サブエリアを指し示すような構成になっている。

【0084】ステップS1506では、領域登録処理に渡された矩形領域の値を対象サブエリア内の領域情報格納領域に登録する。登録する際は、既に実施形態1でも説明したように、既に格納されている領域情報との比較処理によって、左上の座標値がより小さい方と、右下の座標値がより大きな方の値を求め、その値で変数を更新する処理を行う。

【0085】次に本発明の領域登録処理の中で使用されるエリア結合処理について詳細に説明を加える。図16は、領域判定処理の処理の流れを示すフローチャートであり、S1601～S1608は各ステップを示す。エリア結合処理は、複数のサブエリアを結合し、一つ辺りの管理領域を広げるとともに、サブエリア管理用の変数に親と子の従属関係を規定し、管理する変数の数を減らす処理を行う部分である。

【0086】ステップS1601では、引数として渡されてきた新たな矩形領域の左上の座標値より、その点が属するサブエリアのインデックス番号n1を求める。より具体的に説明するために図20、図21を用いて説明を行う。

【0087】図20、図21は、バンド領域内のサブエリア $\Phi$ 、 $\Theta$ 、 $\Psi$ 、 $\Omega$ の描画イメージと、それらを管理するサブエリア管理用配列変数の状態を示すものである。ここで、図20は、エリア結合処理が行われる前の状態、図21は、これから述べるエリア結合処理が施された後の状態を示すものである。図20のような状態に対して、図21に示すような新たな描画オブジェクトとして塗りつぶしパターンが無い四角形を描画する場合を例にとって説明を続ける。図16のフローチャートにおけ

るステップS1601の処理を図21に当てはめて考えると、ここで得られるn1という値は、具体的には、サブエリア $\Phi$ を示すものであり、n1=1ということになる。

【0088】次にステップS1602に進み、そのサブエリアn1内での領域比較を行う。図21の場合では、既にサブエリア $\Phi$ には、直線が描画されており、この領域情報がChkArea[0]の領域情報登録領域にX1s, Y1s, X1e, Y1eとして登録されている。この領域情報と、新たな描画オブジェクトの左上の点Aの領域情報X1e, Y1eとを比較すると、既に登録してある領域情報の方がより左上で小さな値であるため、この場合は、更新は行わない。このようにしてステップS1603では、サブエリア内での領域比較を行い、より小さい左上の座標値をもって更新作業を行う。

【0089】次にステップS1603では、そのサブエリアを親と規定する。図21では、ChkArea[0]のflagが2となり(図17の1702)、結合したことを意味するとともに、parentAreaの番号(図17の1701)が0(サブエリア $\Phi$ の配列のインデックス番号)となり自分自身が親であることを示す。次にステップS1604では、新たな描画オブジェクトの矩形領域情報の右下の座標点を見て、その点が属するサブエリアを特定し、そのインデックス番号をn2とする。n2が求まった時点で、親エリアの管理用変数ChkRect[n1]のサブエリア構造体refRect.right(図17の1705)とrefRect.bottom(1706)を更新する。

【0090】図21の場合は、サブエリア $\Phi$ がn2に該当するため、n2=7となり、ChkRect[0]のrefRect.right(図17の1705)とrefRect.bottom(図17の1706)は、それぞれX2, Y2となる。次にステップS1605において、右下の座標点の領域比較を行う。より右下に大きな値を持つ方の値で領域情報を更新するが、その際は、親となっているサブエリアn1の変数領域の情報を更新する。

【0091】つまり図21においては、サブエリア $\Phi$ を管理しているChkArea[0]のregRectの情報のrightとbottomの値が更新されることになる。つまりChkArea[0]のregRect.right(図17の1709)には、X7sが、regRect.bottom(図17の1710)には、Y7sが格納される。

【0092】ここまでで親となるサブエリアの管理用変数Chkarea[n1]の値は更新されたことになるが、以降では、従属する子サブエリアの管理用変数の更新作業を行う。

【0093】まず、従属する範囲を求めるため、ステップS1606においてn2-n1を計算し、その差が5以

上かどうかで処理を切り分ける。この結果が5以上になる場合は(ステップS1606-Yes)、バンド内のサブエリアがY方向に対して2段にまたがることを示す。

【0094】このような場合は、ステップS1607において、サブエリア(n1+1)から(n2-5)までと、サブエリア(n1+5)からn2までのChkRect[]のflagを2(「結合」であることを識別表示する)とし、さらにparentAreaの番号を親エリアの管理変数の配列のインデックス値である0に更新する。こうすることによってサブエリアに親と子の従属関係を築き、そのような関係にあるサブエリアは新たなサブエリアとして認識するように動作するものである。

【0095】次に本発明の実施形態2における領域判定処理について詳細に説明を加える。

【0096】図22は、領域判定処理の流れを示すフローチャートであり、S2201～S2211は各ステップを示す。まず、対象とする文字列の外接矩形領域がどのサブエリアと関係するのかを調べる。ステップS2201では、まず文字列の外接矩形領域の左上の値が属するサブエリアのインデックス値n1を求める。次にステップS2202において、文字列の外接矩形領域の右下の値が属するサブエリアのインデックス値n2を求める。

【0097】次にステップS2203においてn2-n1を計算し、その結果が5以上かどうかで処理を切り分ける。この結果が5以上になる場合は、バンド内のサブエリアがY方向に対して2段にまたがっていることを示す。このような場合は、ステップS2204、S2205においてまず1段目のX方向のサブエリアn1から(n2-5)に対してそれぞれの管理領域に登録されている領域情報と比較を行う。次に2段目のX方向のサブエリア(n1+5)からn2に対してそれぞれの管理領域に登録されている領域情報と比較を行う(ステップS2206、S2207)。ステップS2205、S2207、S2209で用いる重なり判定ルーチンについては後述する。

【0098】一方ステップS2203において、検査領域が2段にまたがらない場合(ステップS2203-No)は、ステップS2208に進み、サブエリアn1からn2までについて、重なり判定を行う(ステップS2209)。ステップS2205、S2207、S2209のそれぞれの重なり判定の結果、ステップS2210、S2211において、重なり有るか、重なり無しかが領域判定処理の戻り値として返されることになる。

【0099】最後に本発明の実施形態2における領域判定処理の中で使用される重なり判定処理について詳細に説明を加える。

【0100】図23は、重なり判定サブルーチンの処理の流れを示すフローチャートであり、S2301～S2

311は各ステップを示す。図中におけるIX、IYは、重なり判定ルーチンの中で用いられる変数であり、N1、N2は、重なり判定ルーチン呼び出す側が設定する引数を示している。まずステップS2301ではIXにN1を代入する。次にステップS2302において、サブエリア管理用配列であるChkArea[IX]のflagの値を見て、その値が2、つまり結合されたサブエリアであれば、ステップS2303に進み、そうでない場合は、ステップS2304に進み、おなじくflagの値を見て、その値が1、つまり独立した単独のサブエリアである場合は、ステップS2305に進み、そうでない場合、つまり、そのサブエリアに対して、一度も描画がされなかった場合のときは、ステップS2307においてIXの値をインクリメントし、次のサブエリアを検索する。

【0101】ステップS2308においては、対象とするサブエリアN1からサブエリアN2までのすべてについて検査したかどうかをチェックする。ステップS2308において、すべての検査が終了した場合は、どの領域とも重なりが無いと判定する(ステップS2310)。

【0102】ステップS2303では、変数IYに親エリアのエンデックス番号を代入する。またステップS2305では、変数IYにIXを代入する。

【0103】ステップS2306において、チェック対象となるサブエリアの管理用配列であるChkArea[IY]の領域情報であるregRectの値を取得し、文字列の外接矩形領域との重なりを比較する。このステップS2309における重なり比較は、実施形態1にて前述した図10でプログラムとして示した方法と同じである。この比較の結果、重なり無しと判定された場合は、ステップS2307に戻り、次のサブエリアについて検証する。比較の結果、重なり有りと判定された場合は、その結果を戻り値として返す。

【0104】以上、述べた方法により、高精度に下地の文字列を切り出すことが可能となる。

【0105】＜実施形態3＞前述の実施形態1、2では、通常、文字の下地となる網掛けパターンやスプレッドシートの罫線は文字を取り囲むように存在するため、DDIとして情報が渡ってくる場合も文字の方が後になるという前提条件のもとに成り立ち、印刷処理速度を向上させる効果をうたったが、本実施形態では、そのような前提条件が無い場合でも本発明が適用可能であることを示す。

【0106】例えば、図24に示すように、まず始めに⊙として矩形(太さのある直線)の描画情報が入力され、その次に⊙として文字列“ABC”が入力され、最後に⊙として下地との透過度指定付きの楕円描画が入力されたとする。⊙の矩形と⊙の文字列は、それぞれ領域的には重なりが無いため、本発明によれば、⊙の文字列

はPDLとして処理される。つまり④の矩形の描画はホスト側の描画処理実行部によってホスト側の描画メモリに展開されるが、⑤の文字列は、プリンタ側の描画処理実行部によってプリンタ側の描画メモリで印刷処理されることになる。ここで、次に⑤の描画オブジェクトが入力された場合、実施形態1、2では、⑤の描画オブジェクトはホスト側の描画メモリに展開され、⑤の描画が終わった段階でそのバンド内の描画オブジェクトが他にないとすると、④と⑤の描画が行われた描画メモリがプリンタ側にイメージとして送られ、そのイメージの上に⑥の文字列が上書き印刷処理されることになる。この結果、⑤の描画オブジェクトの透過度指定が意味を持たなくなってしまう、正しい印刷結果が得られないこととなる。

【0107】つまり、文字列⑥の描画領域に対して、それより後にその領域を参照するような描画命令が来た場合は、印刷結果の正当性がくずれる可能性があったということである。この問題に対して、本実施形態では、2つの手段によって問題解決が可能なことを示す。

【0108】1つの問題解決手段は、PDLとして処理した文字の領域情報も別途管理し、文字より後に描画される描画オブジェクトがそのエリアと重なるかどうかを判断することである。PDLとして処理した文字列の領域と重なりと判明した時点で、次のいずれかの方法によって、画像の正当性は保証されることになる。

【0109】<方法1>PDLとして処理した文字列の領域と重なりと判明した時点で、画像保証フラグというフラグを立て、それまでにホスト側の描画処理実行部で処理された描画メモリをPDLのコマンドを使ってイメージとしてプリンタ側に送り、画像保証フラグが立った以降は、本発明の判別処理はスキップし、以降の処理はすべてPDLで行うようにする。この方法では、最適なパフォーマンスは出ない可能性があるが、画像の正当性（描画順序の繰り上がりによる印刷結果の不整合がないこと）は保証されることになる。

【0110】方法1を図24の印刷に当てはめてみると、⑤の描画オブジェクトを処理する時点で、PDLとして処理した文字列⑥の領域と重なりとが判明し、まずそれまでにホスト側の描画メモリに展開されたものとして④の矩形が描画された描画メモリの内容をイメージとしてプリンタにPDLのイメージ描画コマンドを使って送り、続く⑥の処理以降はすべてPDLとして処理することになる。

【0111】<方法2>そのバンド内で、PDLとして処理した文字列の印刷情報（書体や文字コード、印刷位置）は、ホスト側のメモリなどに一時的に保持しておく。PDLとして処理した文字列の領域と重なりと判明した時点で、一時的に保持している文字列の印刷情報をメモリから呼び出し、その情報にしたがって、ホスト側の描画処理実行部を使って、ホスト側の描画メモリに展

開する。

【0112】この段階で、領域管理の情報を、文字印刷の領域自体も対象として更新する。こうすることによりそれまでの描画順序に従った正しい印刷イメージがホスト側の描画メモリに形成されることになる。一時的に保持した文字列印刷情報はこの段階で破棄され、再びプリンタドライバとしての通常の処理を継続することが可能である。ホスト側で処理した印刷イメージをプリンタ側に送出する場合は、下地になるような描画や印刷があっても、それを消すようにするため、上書き描画を指定してイメージを送るようにする。

【0113】方法2を図24の印刷に当てはめてみると、⑤の描画オブジェクトを処理する時点で、PDLとして処理した文字列⑥の領域と重なりとが判明し、まずそれまでにホスト側のメモリに一時的に格納された文字列⑥の印刷情報を呼び出し、ホスト側の描画メモリに文字印刷処理を行う。続いて領域登録手段により文字列の領域情報で管理されている領域情報を更新する。続いて、⑤の描画オブジェクトをホスト側の描画メモリに展開し、すべての描画処理が終了したので、ホスト側の描画メモリの内容をイメージデータとしてプリンタに送り印刷を行う。

【0114】この際、イメージデータは上書きで描画を行うように指定する。プリンタ側では、すでに文字列⑥の情報がPDLのデータとして入力されており、その処理をした後、ホスト側で展開されたイメージによって文字を上書きして印刷イメージを形成するものである。

【0115】以上述べた方法1、2によって、本発明においてでも画像の正当性を維持することが可能となる。

【0116】<実施形態4>本実施形態では文字とそれ以外の描画オブジェクトを区別することによって、文字印刷処理の効率化を図ることを提案しているが、文字ではなく、グラフィック系のオブジェクトとそれ以外の描画オブジェクトとして区別しても良い。その際は、グラフィックスの処理がホスト側とプリンタ側で適切に分散されることになる。本実施形態では処理の高速化を図るためにその描画オブジェクトを含む矩形領域によって処理を行っているが、ポリゴンによって描画されるような任意の多角形に対して厳密に領域判断を行うようにしても良い。

【0117】本実施形態では、PDLタイプの処理時は、描画処理実行部としてプリンタ内の描画処理部を想定したが、これは別のパーソナルコンピュータ上で動作するRIP（Raster Image Processor）と呼ばれるモジュールをネットワーク経由で用いてデータ処理するような構成であってもかまわない。ホストコンピュータ側の処理能力が十分に高い場合や、プリンタ側に描画処理実行部を持たないようなプリンタで印刷する場合は、このような構成にすることによって本発明を実現可能となる。

【0118】実施形態2では、バンド内の領域を図13に示すような10個の領域に分けて領域管理を行ったが、領域の分割数をアプリケーションに応じて適応的に変化させることで、画像に応じて判定領域の細かさが決まるため、判定の厳密さが必要に応じて高くなるとともに、パフォーマンスも最適なレベルにチューニング可能となる。例えば、文字が多用される表計算ソフトでは細かく、文字があまり使われないDRAW系では荒くすることで、適応的にパフォーマンスがチューニングされる。

【0119】

【他の実施形態】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェース機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0120】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0121】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0122】プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0123】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0124】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0125】本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説

明すると、図25のメモリマップ例に示す各モジュールを記憶媒体に格納することになる。すなわち、少なくとも「印刷情報生成モジュール2501」「領域情報登録モジュール2502」および「領域判定モジュール2503」の各モジュールのプログラムコードを記憶媒体に格納すればよい。

【0126】

【発明の効果】以上、説明したように、各DDIの情報をスプールすることなく逐次的に処理し、文字以外（DrvTextOut以外）の描画オブジェクトの最大領域をその都度求め、その領域を事前に区切ったいくつかのサブエリアとして管理し、DrvTextOutにおいて、その領域とのオーバーラップをチェックし、オーバーラップがある部分の文字はイメージタイプの描画処理（ホスト側の描画メモリで描画処理の実行を行う）として、オーバーラップがない部分の文字はPDLタイプの処理（プリンタ側の描画メモリで描画処理の実行を行う）としてプリンタに送出することで高速な印刷処理を実現するものである。

【0127】通常、文字の下地となる網掛けパターンやスプレッドシートの罫線は文字を取り囲むように存在するため、DDIとして情報が渡ってくる場合も文字の方が後にあるであろうという前提条件のもとに本実施形態1、2は成り立っている。

【0128】但し、実施形態3で示したように、この前提が崩れた場合においても、印刷パフォーマンスは若干低下する可能性があるものの、印刷結果の正当性は保証できる。

【0129】このようなイメージタイプとPDLタイプの両者の良い部分の要素を適切に切り替えられるような構成にすることによって、

○イメージタイプであっても、下地のない文字として切り出せた場合は、バンド間にまたがる文字の二重印刷解消が可能となる。

【0130】○下地が無い領域の文字のみ分離可能であり、イメージと文字の印刷が従来のイメージタイプ、PDLタイプのユーザによる切り替え型のドライバに比べ、切り替える負担をなくし、かつ適切なパフォーマンスが得られるようになる。

【0131】といった効果が期待できるものである。

【0132】特に、複雑度の高いドロー系に強いイメージタイプのドライバと文字印刷に強いPDLタイプのドライバの両方の良さを兼ね備えたバランスの良いドライバであり、従来より高速な印刷処理を実現可能といった効果が期待できる。

【0133】

【図面の簡単な説明】

【図1】本発明の実施形態が適用可能なレーザビームプリンタの構造を示す側断面図である。

【図2】本発明の実施形態である印刷システム全体の基

本構成を示すブロック図である。

【図3】従来の課題について説明する印刷イメージの一例および生成コードの違いを示す概念図である。

【図4】プリンタドライバのユーザインターフェースの一例を示す図である。

【図5】印刷イメージの一例を示す図である。

【図6】バンドにまたがる文字印刷時の処理説明のための概念図である。

【図7】バンドに展開される描画オブジェクトの模式図である。

【図8】処理の流れを示す概念図である。

【図9】領域登録処理のC言語によるプログラム例を示す図である。

【図10】領域判定処理のC言語によるプログラム例を示す図である。

【図11】文字印刷DDI処理のC言語によるプログラム例を示す図である。

【図12】実施形態1で効果が期待できない印刷イメージの一例を示す図である。

【図13】実施形態2における分割管理されるバンド領域の図である。

【図14】プリンタドライバの処理の流れを示すフローチャートである。

【図15】実施形態2の領域登録処理の流れを示すフローチャートである。

【図16】実施形態2のエリア結合処理の流れを示すフローチャートである。

【図17】実施形態2で用いるSUBAREA構造体の型宣言を示す図である。

【図18】実施形態2におけるサブエリア管理用配列の初期状態の模式図である。

【図19】実施形態2におけるサブエリア $\Phi$ に収まる場合の管理用配列の状態を示す図である。

【図20】実施形態2におけるサブエリア結合処理前の管理用配列の状態を示す図である。

【図21】実施形態2におけるサブエリア結合処理後の管理用配列の状態を示す図である。

【図22】実施形態2の領域判定処理の流れを示すフローチャートである。

【図23】実施形態2の重なり判定処理の流れを示すフローチャートである。

【図24】実施形態3における、描画順序の繰り上がりによる印刷結果の不整合解消を説明する図である。

【図25】記録媒体のメモリマップを説明する図である。

#### 【符号の説明】

|      |                |
|------|----------------|
| 100  | 印刷装置(LBP)      |
| 110  | フォーマット制御部      |
| 111  | インタフェース        |
| 1121 | 受信バッファ         |
| 1122 | 送信バッファ         |
| 113  | PDL解析部         |
| 114  | 印刷制御処理実行部      |
| 115  | 描画処理実行部        |
| 116  | ページメモリ         |
| 120  | オペレーション・パネル    |
| 130  | 出力制御部          |
| 131  | レーザドライバ        |
| 140  | プリンタエンジン部      |
| 141  | 半導体レーザ         |
| 142  | レーザ光           |
| 143  | ポリゴンミラー        |
| 144  | 静電ドラム          |
| 145  | 現像ユニット         |
| 146  | 給紙カセット         |
| 147  | 給紙ローラ          |
| 148  | 搬送ローラ          |
| 149  | 搬送ローラ          |
| 200  | ホスト・コンピュータ     |
| 201  | アプリケーション・ソフト   |
| 202  | グラフィック・サブ・システム |
| 2021 | GDI            |
| 2022 | プリンタ・ドライバ      |
| 203  | スプール・サブ・システム   |
| 204  | インタフェース        |
| 210  | キーボード          |
| 211  | マウス            |
| 220  | ディスプレイ・モニタ     |

【図5】



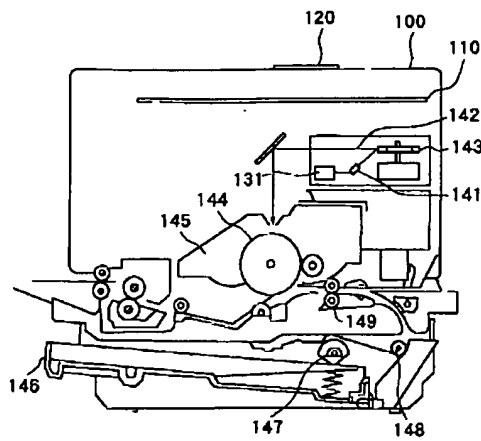
【図9】

```

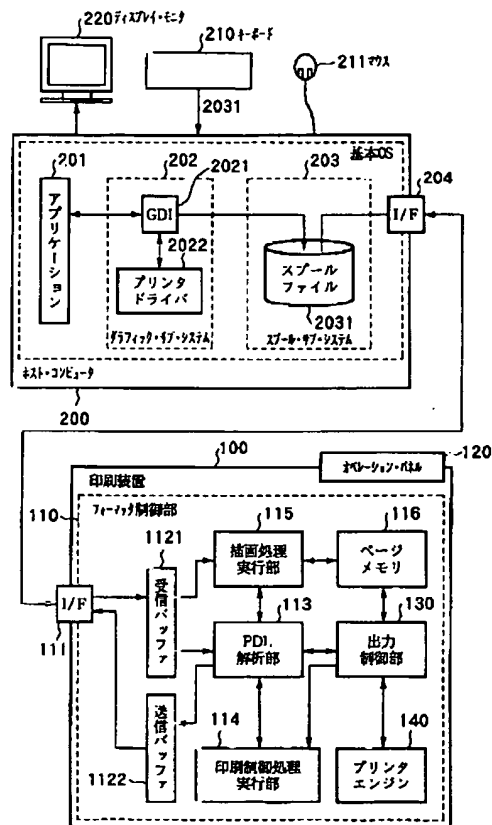
/###領域登録処理###/
ChkRECT.left =min(TrgRECT.left , ChkRECT.left );
ChkRECT.top =min(TrgRECT.top , ChkRECT.top );
ChkRECT.right =max(TrgRECT.right , ChkRECT.right );
ChkRECT.bottom =max(TrgRECT.bottom , ChkRECT.bottom);

```

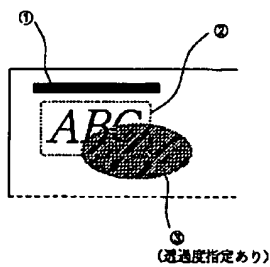
【図1】



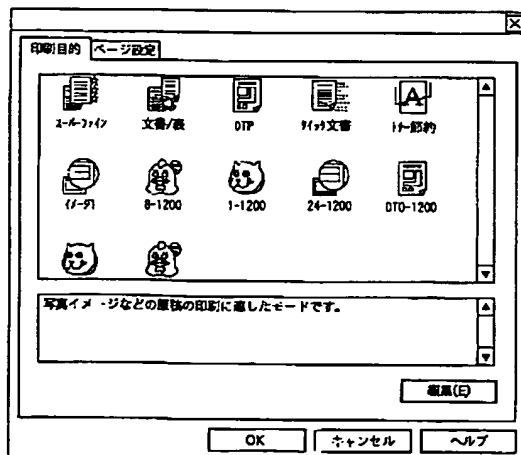
【図2】



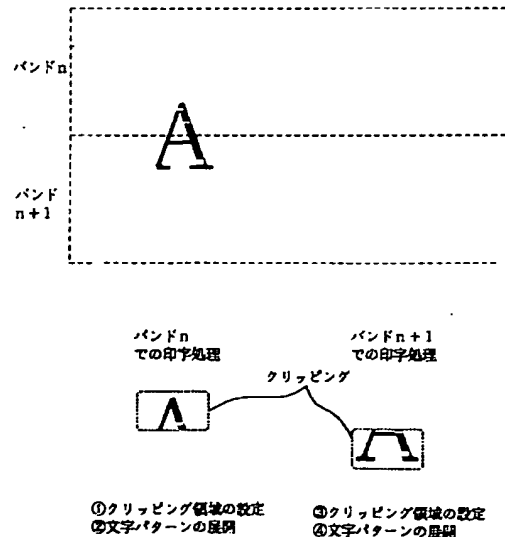
【図24】



【図4】



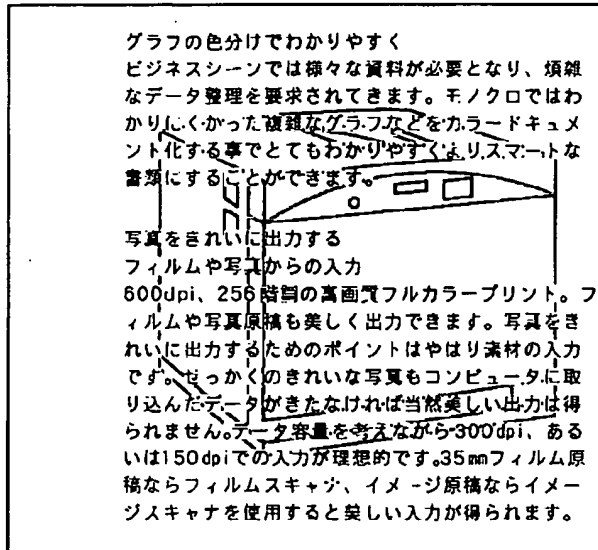
【図6】



①クリッピング領域の設定  
②文字パターンの展開

③クリッピング領域の設定  
④文字パターンの展開

【図3】



【図11】

```

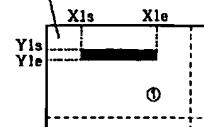
/****文字印字DDI処理****/
BOOL DrvTextOut( . . . . . ){
:
(途中省略)
:
if(bNonOverlap==TRUE){
    ImgTextOut( . . . . . );
}
else{
    PDLTextOut( . . . . . );
}
}

```

【図19】

ChkArea[0]

| 0   | 1 |
|-----|---|
| X0  |   |
| Y0  |   |
| X1  |   |
| Y1  |   |
| X1s |   |
| Y1s |   |
| X1e |   |
| Y1e |   |



イメージタイプの  
プリンタドライバの  
生成コード

ホスト側で展開した  
印刷イメージの  
データ量

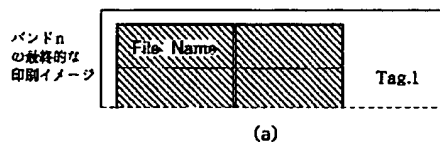
PDLタイプの  
プリンタドライバの  
生成コード

ホスト側で展開した  
印刷イメージの  
データ量

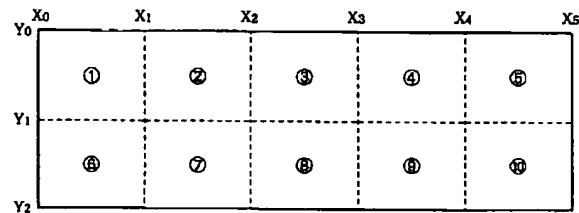
+

分離処理した  
文字情報の  
データ量

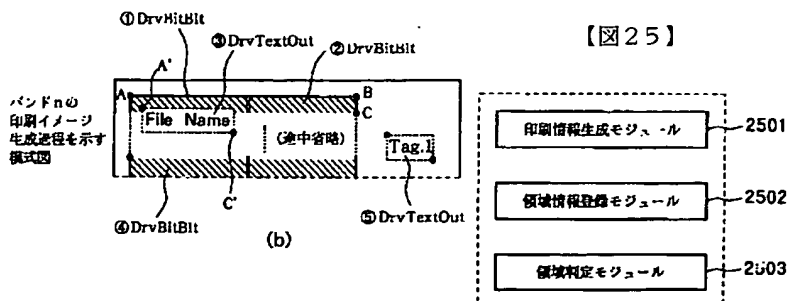
【図7】



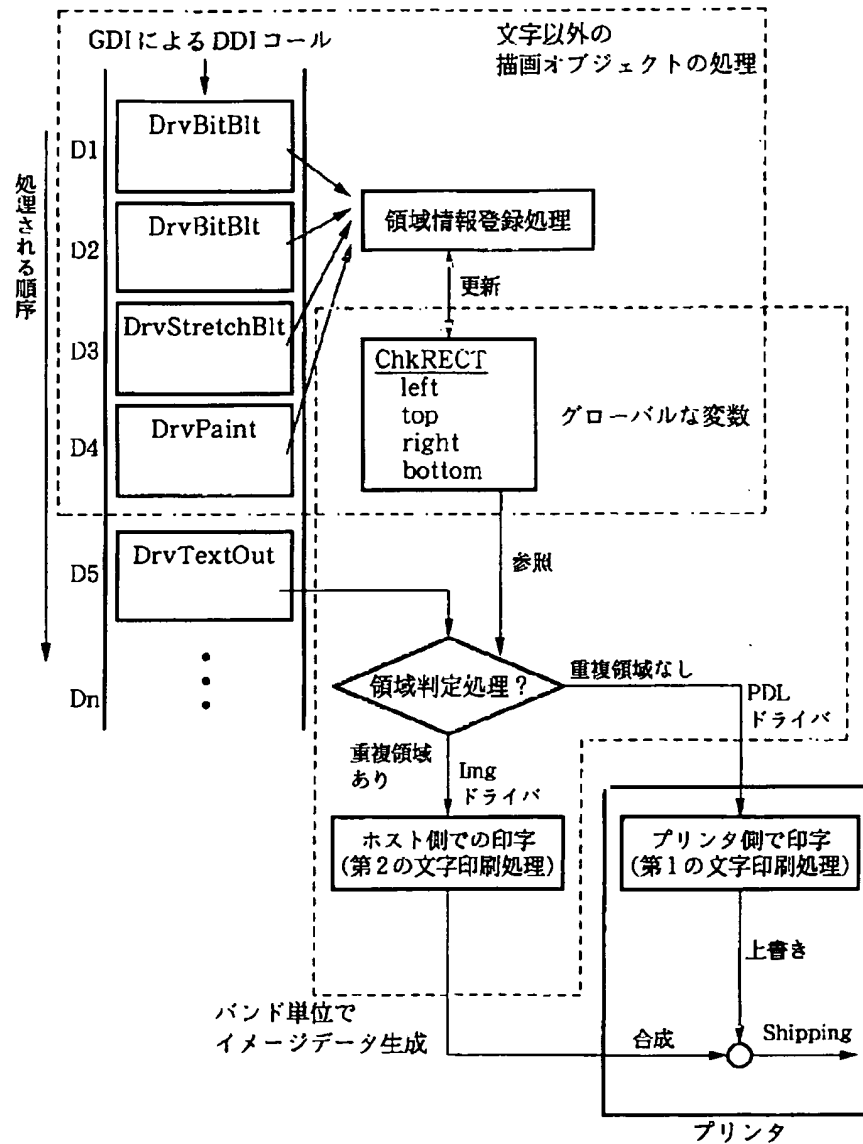
【図13】



【図25】



【図8】



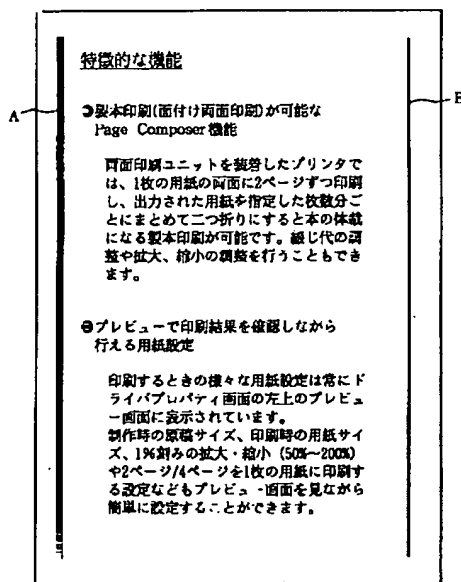
【図10】

```

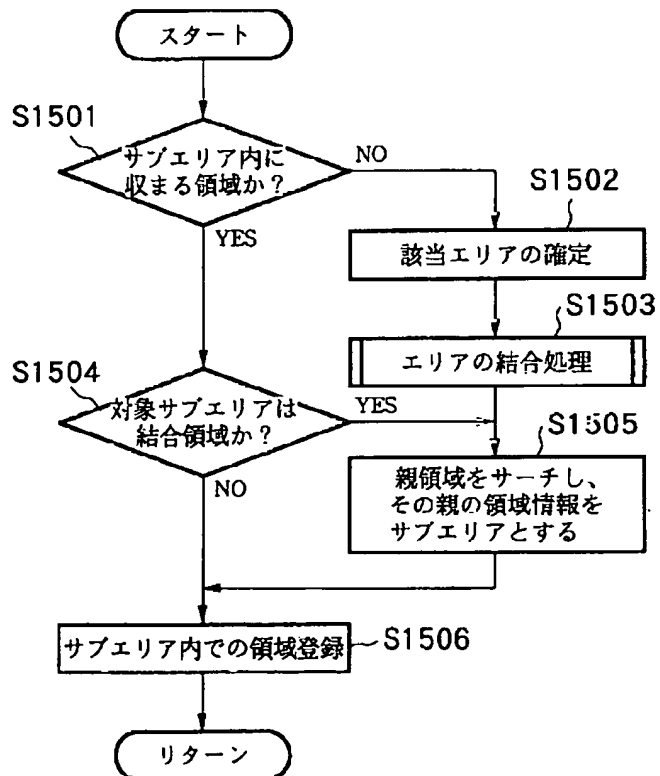
/****領域判定処理****/
BOOL ChkRect( . . . ){
    BOOL bNonOverlap=FALSE;
    if(!((ChkRECT.left==ChkRECT.right)||ChkRECT.top==ChkRECT.bottom)))
        if(pco->rclBounds.right<ChkRECT.left)
            ||(ChkRECT.right<pco->rclBounds.left)
            ||(pco->rclBounds.bottom<ChkRECT.top)
            ||(ChkRECT.bottom<pco->rclBounds.top))
                bNonOverlap=TRUE;
    }
    return(bNonOverlap);
}

```

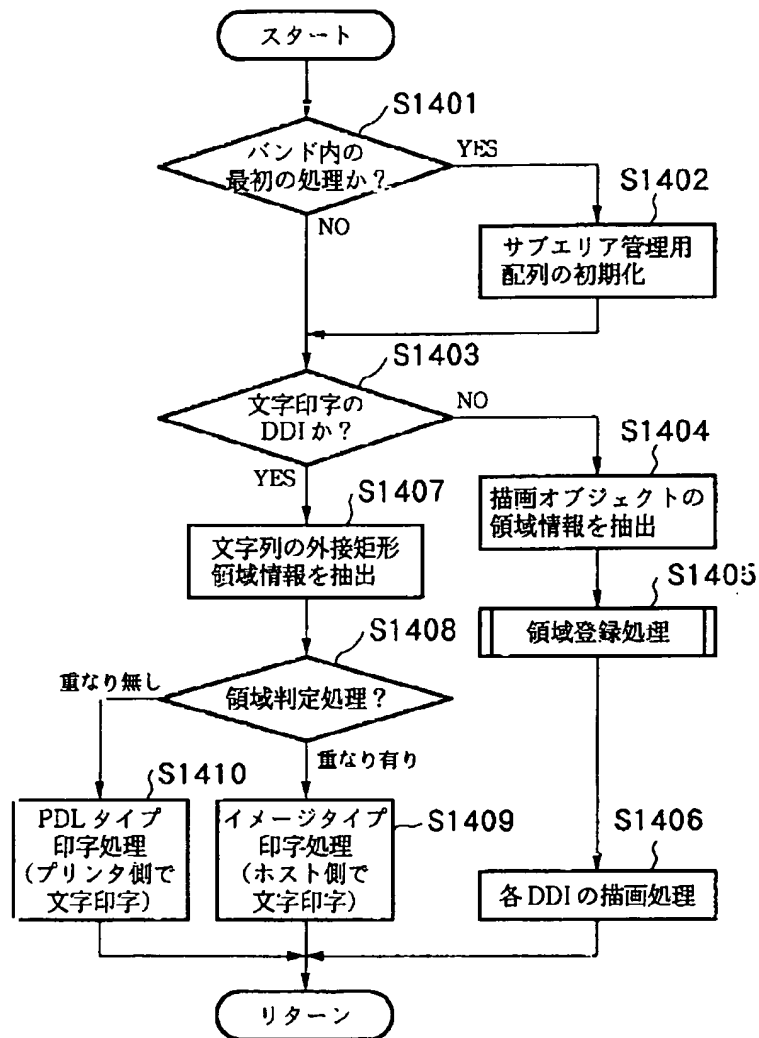
【図12】



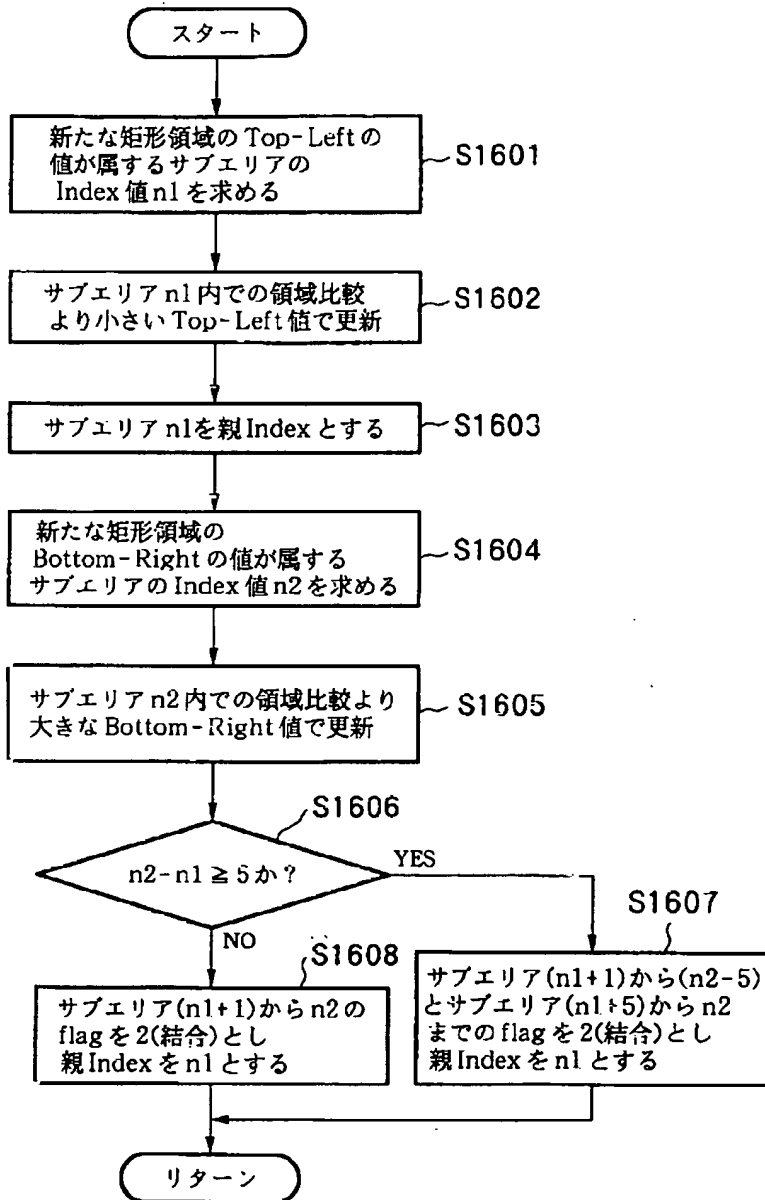
【図15】



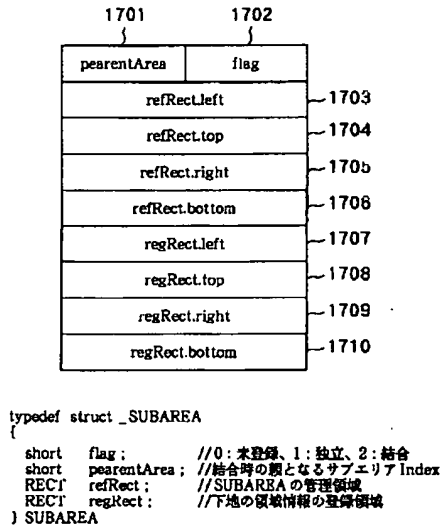
【図14】



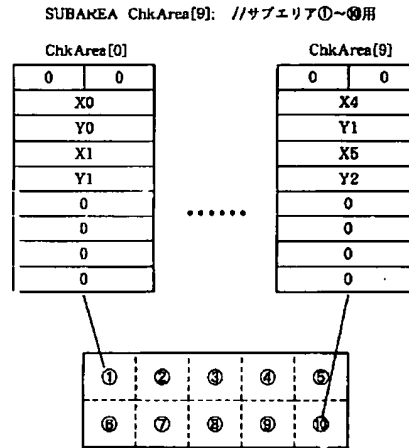
【図16】



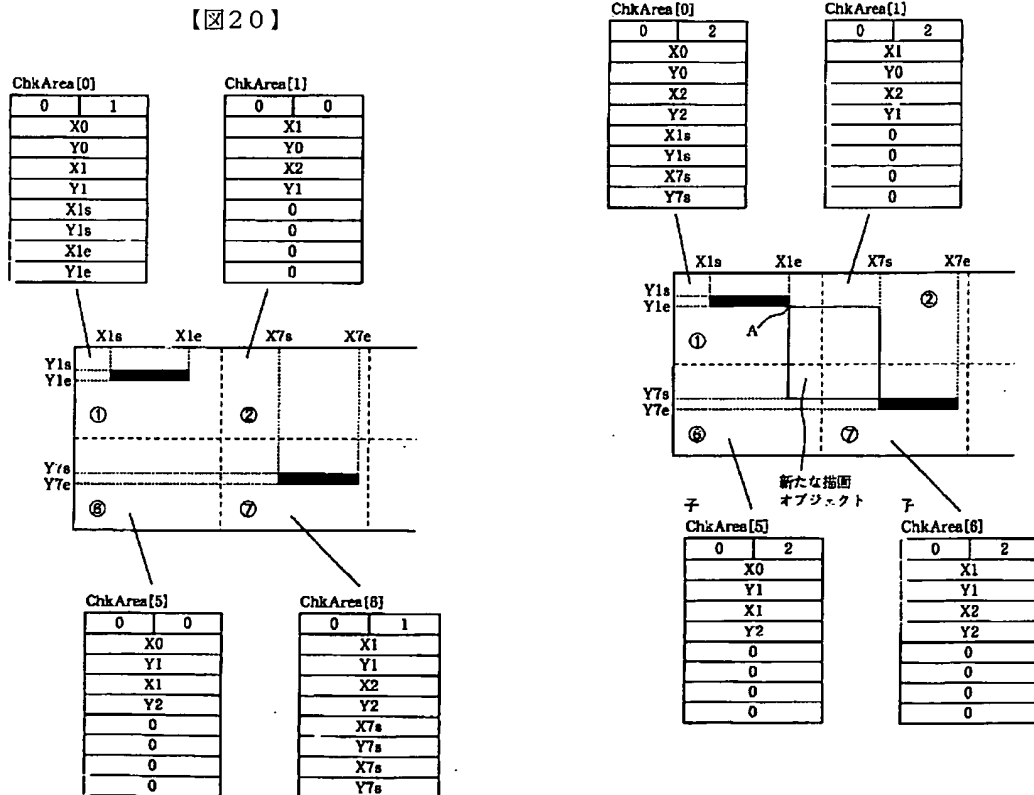
【図17】



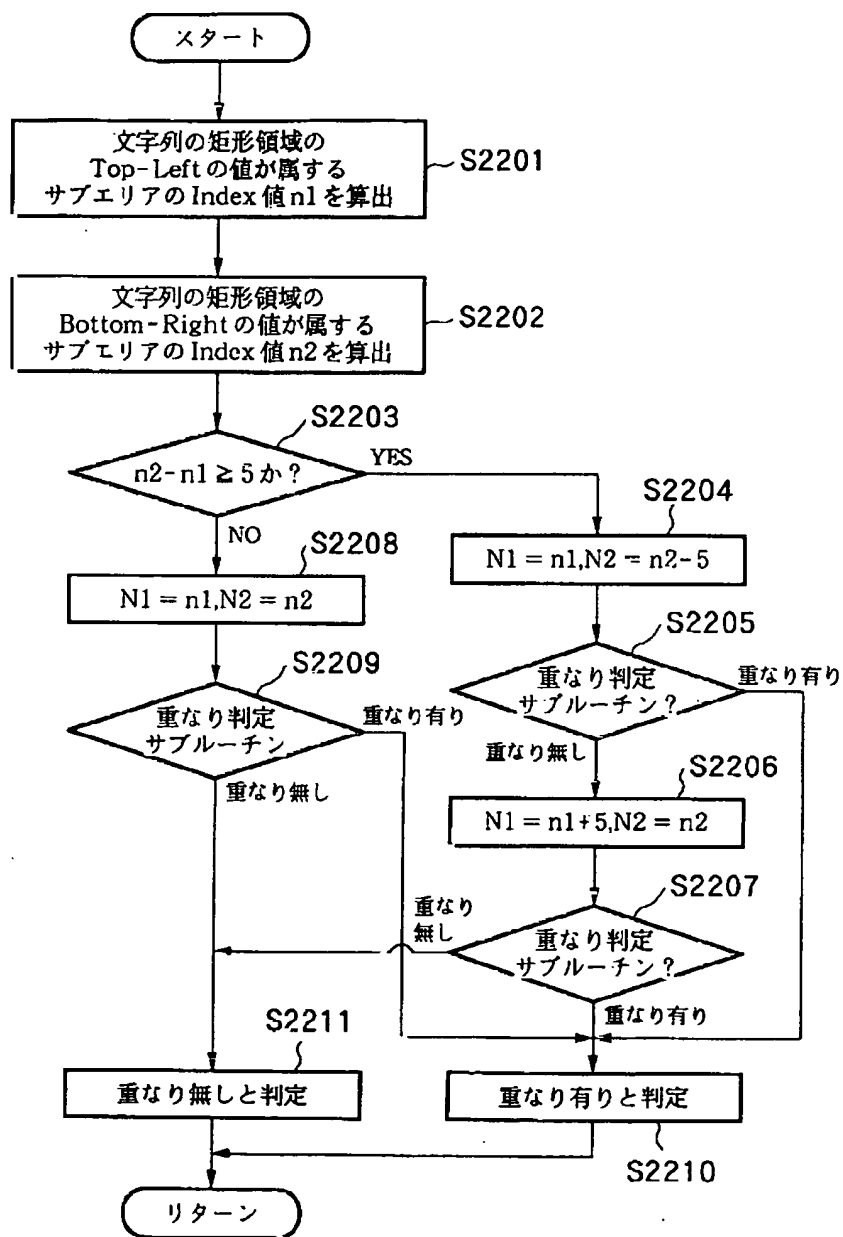
【図18】



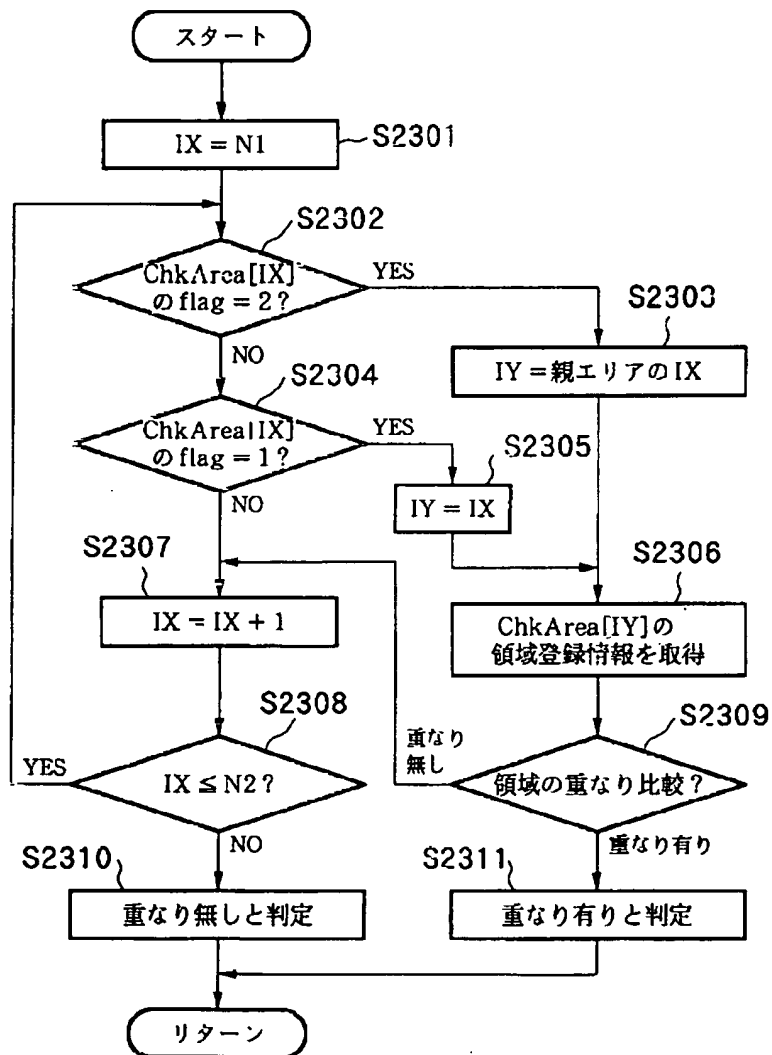
【図21】



【図22】



【図23】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**